INDEX

SYNOPSIS

This project contains the code for an Automobile Management System for a showroom named “PHOENIX AUTOMOBILES”. This Automobile Management System helps in the easier and efficient working of the showroom. It helps in the easier management of the Automobile showroom and reduces the management time.

The project covers the code for both the company/agency point of view and the customer point of view. The program first creates and inputs data into various tables. Later, important functions are implemented as per requirement.

Customers can first compare between the details of 2 vehicles. Then, they can enter the chosen vehicle. They can then choose the Bank along with the rate and time for Emi payment. Then, the customers have to enter their details and the bill would be printed.

Agents/Employees, however, has a bigger control of the program. They can add Employee details, view Employee details, view Customer details, view number of vehicles sold so far, view Ally Companies and view Amount Collected so far along with the Biggest Sale Of the Day.

PYTHON INTRODUCTION

Python is a high-level programming language that is easy-to-learn yet powerful. Python programming language was developed by Guido Van Rossum in February 1991.

Some features of Python programming language are:

- Easy to use
- Dynamic typing
- Interpreted language
- Cross-platform language
- Free and open source

Python provides lots of libraries which allows users to perform wide range of functions. It is also useful in developing various applications.

Many globally known applications make use of python. Some of them are:

- YouTube
- DropBox
- BitTorrent
- Instagram
- Pinterest
- Spotify
- Uber
- Reddit

Some of the applications of python include:

- Web Development
- Machine Learning and Artificial Intelligence
- Data Science and Data Visualization
- Desktop GUI Interface
- Game Development

MYSQL INTRODUCTION

MySQL is a freely available open-source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). In a MySQL database, information is stored in the form of tables. A single MySQL database can contain many tables at once and store thousands of individual records.

MySQL provides you with a rich set of features that supports a secure environment for storing, maintaining, and accessing data. MySQL is a fast, reliable, scalable alternative to many of the commercial RDBMSs available today.

MySQL is part of LAMP (Linux, Apache, MySQL, PHP/Perl/Python) environment, a fast-growing open-source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost, reliability and documentation.

Some key features of MySQL are:

- Speed
- Ease of use
- Free of cost
- Query Language Support (SQL)
- Portability
- Security
- Connectivity

Applications that use MySQL:

- Facebook
- Twitter
- YouTube

PYTHON-MYSQL CONNECTOR INTRODUCTION

When you design real life applications, you are bound to encounter situations wherein you need to manipulate data stored in a database through an application designed by you.

Python MySQL Connector is a Python library that helps to integrate Python and MySQL. This Python MySQL library allows the conversion between Python and MySQL data types. It enables Python programs to access MySQL databases. It is written in pure Python and does not have any dependencies except for the Python Standard Library.

There are mainly seven steps that must be followed in order to create a database connectivity application:

1) Start Python.
2) Import the packages required for database programming.
3) Open a connection to database.
4) Create a cursor instance.
5) Execute a query.
6) Extract data from result set.
7) Clean up the environment.

The library that has to be imported is mysql.connector.

PROGRAM CODE

CONTENTS OF THE PROGRAM

Tables:

Bikes : To store the details of Bikes available
Scooter : To store the details of Scooters available
Cars : To store the details of Cars available
Bank : To store the details of Bank along with rate provided (for Emi)
Ally_Companies : To store the details of Ally Companies
Customers : To store the details of Customers
Employee : To store the details of Employees

Functions:

employee() : To input employee details
agency_pov() : Deals with functions related to agency point of view.
Performs the functions:

1) To add Employee details
2) To view Employee details
3) To view Customer details
4) To view number of vehicles sold so far
5) To view Ally Companies
6) To view Amount Collected so far with the Biggest Sale Of the Day

Comparison() : To compare the details of inputted 2 vehicles
Customers() : To insert data into Customers table
updateagent() : To update the number of vehicles sold for entered agent in Employee table
EMIfunc() : To calculate the Emi
Customerpov() : Deals with functions related to customer point of view. The customer can compare vehicles and place the order for buying a vehicle. Finally, the BILL is printed.

*Note- The entire program is run together.
*A database called ‘project’ was created in MySQL and the program uses it.

PROGRAM

```
import mysql.connector
con=mysql.connector.connect(host="localhost",user='root',passwd=",database='project')
vehCur=con.cursor()

#TO CREATE TABLE BIKES

vehCur.execute('create table bikes(Sl_No int,MODEL  char(30) primary key,COLOURS_AVAILABLE
char(50),FUEL_TYPE varchar(20),BODY_TYPE varchar(30),MILEAGE float, TRANSMISSION
varchar(20),ENGINE_TYPE char(50),FUEL_TANK_CAPACITY float,No_Of_CYLINDERS
int,KERB_WEIGHT float,DISPLACEMENT float,EMISSION_STANDARD char(25),PRICE
float,WARRANTY char(25),STOCK_LEFT int)')

vehCur.execute("insert into bikes values(1,'APACHE RR
310','BLACK,RED,MAROON','PETROL','SPORTS BIKE',35,'MANUAL','SINGLE CYLINDER,4-
STROKE,4-VALUE,REVERSE INCLINED DOHC S1 ENGINE',11,1,174,312.2,'bs6',219000,5,10)")
vehCur.execute("insert into bikes values(2,'APACHE RTR
160','BLUE,RED,BLACK,GREY,MAROON,WHITE','PETROL','SPORTS NAKED
BIKES',47,'MANUAL','S1,4-STROKE,AIR-COOLED',12,1,174,159.7,'bs6',130000,5,20)")
vehCur.execute("insert into bikes values(3,'TVS STAR CITY
PLUS','RED,WHITE','PETROL','COMMUTER BIKES',65,'MANUAL','SINGLE LINE,4
STROKE,AIR COOLED',10,1,116,109.7,'bs6',84561,5,100)")
vehCur.execute("insert into bikes values(4,'BAJAJ PULSAR F250','RED,GREY','PETROL','SPORTS
BIKE',39,'MANUAL','SINGLE CYLINDER 4 STROKE SOHC2 VALVE OIL
COOLED',14,1,162,249.07,'bs6',165000,5,50)")
vehCur.execute("insert into bikes values(5,'ROYAL ENFIELD BULLET
350','BLACK,RED,BLUE','PETROL','CRUISER BIKES',35,'MANUAL','SINGLE CYLINDER 4
STROKE,AIR COOLED FUEL INJECTION',13.5,1,191,346,'bs6',182000,5,20)")
vehCur.execute("insert into bikes values(6,'HONDA HORNET
2.0','BLUE,BLACK','PETROL','SPORTS NAKED BIKES',45,'MANUAL','4 STROKE,SL
ENGINE,BS VI',12,1,142,184.4,'bs6',152000,6,100)")
vehCur.execute("insert into bikes values(7,'HONDA
UNICORN','BLACK,GREY,RED','PETROL','COMMUTER BIKES',60,'MANUAL','4
STROKE,S1,BS-VI ENGINE',13,1,193,162.7,'bs6',177000,6,350)")
vehCur.execute("insert into bikes values(8,'ROYAL ENFIELD HIMALAYAN','MIRAGE
SILVER','PETROL','ADVENTURE TOURER BIKES',36,'MANUAL','SINGLE CYLINDER,4
STROKE,AIR COOLED,SOHC',15.5,1,195,411,'bs6',242000,5,5)")
vehCur.execute("insert into bikes values(9,'HARLEY DAVIDSON ELECTRA GLIDE
STANDARD','BLACK','PETROL','TOURER BIKES',18,'MANUAL','MILWAKEE-EIGHT R
107',22.7,2,354,1745,'bs6',2779000,2,10)")
vehCur.execute("insert into bikes values(10,'BMW R NINE T
SCRAMBLER','GREY,BROWN,BLUE','PETROL','CAFE RACER BIKES',19.6,'MANUAL','Air/Oil
cooled 2-cylinder,4-stroke,boxer engine with 2 camshafts & 4 radially arranged valves per cylinder as
well as central counter balance shaft',17,2,223,1170,'bs6',1865000,2,10)")
vehCur.execute("insert into bikes values(11,'HERO XTREME 160R','RED','PETROL','SPORTS
NAKED BIKES',19.6,'MANUAL','AIR COOLED,4 STROKE,2 VALVE SINGLE CYLINDER
OHC',12,1,139.5,163,'bs6',134000,5,410)")
```

```
vehCur.execute("insert into bikes values(12,'SUZUKI
INTRUDER','GREY,BLACK','PETROL','CRUISER BIKES',44,'MANUAL','4-STROKE,1-
CYLINDER AIRCOOLED,SOHC',11,1,152,155,'bs6',148000,2,20)")
vehCur.execute("insert into bikes values(13,'DUCCATI STREETFIGHTER
V4','BLACK','PETROL','SUPER BIKES',13,'MANUAL','Desmosedici strodale 90
VA',16,4,199,1103,'bs6',2577000,2,32)")
vehCur.execute("insert into bikes values(14,'KTM 200 DUKE','ORANGE,WHITE','PETROL','SPORTS
NAKED BIKES',35,'MANUAL','SINGLE CYLINDER,4 VALVE LIQUID
COOLED,F1,DOHC',13.5,1,159,199.5,'bs6',210000,2,100)")
vehCur.execute("insert into bikes values(15,'KAWASAKI NINJA ZX-10R','-','PETROL','SUPER
BIKES',15,'MANUAL','LIQUID COOLED, 4-STROKE IN LINE
FOUR',17,4,207,998,'bs6',1713000,2,2)")
con.commit()
```

#TO CREATE TABLE SCOOTER

```
vehCur.execute("CREATE TABLE scooter(Sl_No int,MODEL char(25) primary
key,COLOURS_AVAILABLE char(50),ENGINE_TYPE char(50),RANGE_ESCOOTER
int,MOTOR_POWER int,CHARGING_TIME float,BATTERY_CAPACITY float,CHASSIS
char(25),TOP_SPEED float, MILEAGE float,DISPLACEMENT float,EMISSION_STANDARD
char(20),FUEL_TANK_CAPACITY float,LOAD_CAPACITY int,PRICE float,WARRANTY
int,STOCK_LEFT int)")

vehCur.execute("INSERT INTO scooter VALUES(1,'BMW C 400 GT','Cherry Red,Blue,Pastel
Pink,Black,White','Water-cooled, single-cylinder, 4 stroke engine, 4 valves per cylinder, overhead
camshaft',Null,Null,Null,Null,Null,93.56,53.5,124.5,'BS6',12.8,201,950000,6,30)")
vehCur.execute("INSERT INTO scooter VALUES(2,'Suzuki Burgman Street','Red,Blue,Baby
Pink,Black,White,Dark Green','4-Stroke, 1 Cylinder, Air
Cooled',Null,Null,Null,Null,Null,90.48,46.85,124,'BS6',5.5,110,77500,5,350)")
vehCur.execute("INSERT INTO scooter VALUES(3,'Bajaj Chetak','Cherry Red,Blue,Pastel
Blue,Black,White,Yellow,Grey','Single Cylinder, 4 stroke, Air cooled, 3
valves',Null,Null,Null,Null,Null,70,50.25,125,'BS6',8.4,126,350000,4,42)")
vehCur.execute("INSERT INTO scooter VALUES(4,'Hero Maestro
125','Yellow,Blue,Pink,Black,White','Air Cooled, 4-Stroke, SI
Engine',Null,Null,Null,Null,Null,95.26,45.83,124.6,'BS6',5,122,74300,3,33)")
vehCur.execute("INSERT INTO scooter VALUES(5,'Honda Activa
125','Red,Blue,Yellow,Black,White,Purple,Grey','Fan Cooled, 4 Stroke, BS-VI
Engine',Null,Null,Null,Null,Null,92.46,48.2,124,'BS6',5.3,111,82800,2,30)")
vehCur.execute("INSERT INTO Scooter VALUES(6,'Ather 450X','Charcoal
Black',Null,100,5400,5.45,2.9,'Aluminium Cast',80,Null,Null,Null,Null,108,118000,3,20)")
vehCur.execute("INSERT INTO Scooter VALUES(7,'Simple One STD','Sky
Blue',Null,236,4500,1.08,4.8,'Tubular',105,Null,Null,Null,Null,110,109000,4,25)")
vehCur.execute("INSERT INTO Scooter VALUES(8,'Ola S1','Cherry
Red',Null,121,8500,4.48,2.98,'Tubular',90,Null,Null,Null,Null,121,85099,2,24)")
vehCur.execute("INSERT INTO Scooter VALUES(9,'TVS iQube
Electric','White',Null,75,4400,6,2.7,'Telescopic',78,Null,Null,Null,Null,118,100000,3,18)")
vehCur.execute("INSERT INTO Scooter VALUES(10,'Hero Electric Optima','Cherry
Red',Null,85,250,4.5,30,'Aluminium Cast',25,Null,Null,Null,Null,88,51400,3,30)")
con.commit()
```

#TO CREATE TABLE CARS

```
vehCur.execute("CREATE TABLE Cars (Sl_No int, MODEL varchar(20),COLOURS_AVAILABLE
varchar(100),FUEL_TYPE char(10),FUEL_TANK_CAPACITY_Litres float,EMISSION_STANDARD
char(5), BODY_TYPE varchar(20),MILEAGE_kmpl float, PRICE float, INSURANCE int,
WARRANTY int, STOCK_LEFT int)" )

vehCur.execute("INSERT INTO Cars VALUES(1,'Maruti
Swift','White,Grey,Silver,Red,Orange,Blue','petrol',37.0,'BS6','Hatchback',23.76,982268,43878,5,14)")
vehCur.execute("INSERT INTO Cars VALUES(2,'Maruti
Baleno','White,Grey,Silver,Red,Blue','petrol',37.0,'BS6','Hatchback',19.56,1081968,41518,6,9)")
vehCur.execute("INSERT INTO Cars VALUES(3,'Maruti
Celerio','White,Grey,Red,Black','petrol',32.0,'BS6','Hatchback',26.00,774442,31862,4,10)")
vehCur.execute("INSERT INTO Cars VALUES(4,'Tata
Altroz','White,Grey,Red,Black,Brown','diesel',37.0,'BS6','Hatchback',19.05,1048755,37712,5,8)")
vehCur.execute("INSERT INTO Cars VALUES(5,'Tata
Tiago','White,Grey,Red,Blue,Black','petrol',35.0,'BS6','Hatchback',23.84,818637,30294,4,11)")
vehCur.execute("INSERT INTO Cars VALUES(6,'Hyundai
i20','White,Grey,Red,Black','diesel',37.0,'BS6','Hatchback',20.28,1320806,40763,6,7)")
vehCur.execute("INSERT INTO Cars VALUES(7,'Honda
Amaze','White,Grey,Silver,Black','diesel',35.0,'BS6','Sedan',18.6,1016430,34325,6,5)")
vehCur.execute("INSERT INTO Cars VALUES(8,'Toyota
Fortuner','White,Grey,Silver,Black,Brown','diesel',80.0,'BS6','SUV',8.0,4407401,165296,7,3)")
vehCur.execute("INSERT INTO Cars VALUES(9,'Mahindra
Scorpio','White,Red,Black','diesel',60.0,'BS6','SUV',20.5,1578589,63148,6,5)")
vehCur.execute("INSERT INTO Cars VALUES(10,'MG
Hector','White,Red,Black,Brown,Blue','diesel',60.0,'BS6','SUV',20.0,1823465,67647,4,2)")
vehCur.execute("INSERT INTO Cars VALUES(11,'Hyundai
Creta','White,Red,Orange,Silver,Black','diesel',50.0,'BS6','SUV',21.4,1579089,53018,5,4)")
vehCur.execute("INSERT INTO Cars VALUES(12,'Hyundai
Venue','White,Red,Silver,Black','diesel',45.0,'BS6','SUV',23.7,1236679,44185,5,3)")
vehCur.execute("INSERT INTO Cars VALUES(13,'Kia
Seltos','White,Grey,Silver,Black,Red,Orange,Blue','diesel',50.0,'BS6','SUV',20.6,2137249,65884,4,5)")
vehCur.execute("INSERT INTO Cars VALUES(14,'Tata
Nexon','White,Grey,Red','diesel',44.0,'BS6','SUV',21.5,1568365,47903,4,1)")
vehCur.execute("INSERT INTO Cars VALUES(15,'Tata
Harrier','White,Grey,Black','diesel',50.0,'BS6','SUV',17.0,2515732,87838,7,2)")
vehCur.execute("INSERT INTO Cars VALUES(16,'Mahindra
Bolero','White,Silver','diesel',60.0,'BS6','SUV',16.0,1130629,37103,5,4)")
vehCur.execute("INSERT INTO Cars VALUES(17,'Toyota Innova
Crysta','White,Grey,Red,Black','diesel',55.0,'BS6','MUV',12.0,2182707,94547,5,2)")
vehCur.execute("INSERT INTO Cars VALUES(18,'Mercedes-Benz V-
Class','White,Silver,Black,Blue','diesel',57.0,'BS6','MUV',16.0,8407709,302309,6,1)")
vehCur.execute("INSERT INTO Cars VALUES(19,'Kia
Carnival','White,Silver,Black','diesel',60.0,'BS6','MUV',14.11,3033800,108395,7,1)")
vehCur.execute("INSERT INTO Cars VALUES(20,'Maruti
Ertiga','White,Silver,Black,Red,Blue','diesel',45.0,'BS6','MUV',19.01,910879,41504,5,0)")
```

```
vehCur.execute("INSERT INTO Cars VALUES(21,'Renault
Triber','White,Silver,Blue,Mustard','petrol',40.0,'BS6','MUV',20.0,629520,30100,4,2)")
vehCur.execute("INSERT INTO Cars VALUES(22,'BMW
X5','White,Black,Blue','diesel',80.0,'BS6','SUV',13.38,9044132,323132,5,1)")
vehCur.execute("INSERT INTO Cars VALUES(23,'BMW
Z4','White,Black,Blue,Red','petrol',52.0,'BS6','Convertible',11.29,9549712,347812,6,0)")
vehCur.execute("INSERT INTO Cars VALUES(24,'Porsche
911','White,Black,Orange,Red,Yellow,Blue,Green','petrol',64.0,'BS6','Coupe',18.24,35409595,1216045,
5,0)")
vehCur.execute("INSERT INTO Cars VALUES(25,'Bentley
Continental','Silver,Black,Green,Yellow','petrol',90.0,'BS6','Convertible',9.8,44990236,1537710,5,1)")
vehCur.execute("INSERT INTO Cars VALUES(26,'Mercedes-Benz C-
Class','White,Black,Blue','diesel',66.0,'BS6','Sedan',12.6,6023331,144211,5,1)")
vehCur.execute("INSERT INTO Cars VALUES(27,'Mercedes-Benz E-
Class','White,Silver,Black','diesel',80.0,'BS6','Sedan',16.1,7775740,177081,5,0)")
vehCur.execute("INSERT INTO Cars VALUES(28,'Rolls-Royce
Ghost','White,Silver,Black,Blue','petrol',90.0,'BS6','Sedan',6.33,79853217,2708217,6,0)")
vehCur.execute("INSERT INTO Cars VALUES(29,'Rolls-Royce
Phantom','White,Silver,Black,Red,Blue','petrol',100.0,'BS6','Sedan',9.8,103283890,3494890,6,0)")
vehCur.execute("INSERT INTO Cars VALUES(30,'Rolls-Royce
Dawn','Silver,Black,Red,Blue','petrol',100.0,'BS6','Convertible',9.8,83873185,2843185,6,0)")
vehCur.execute("INSERT INTO Cars VALUES(31,'Audi
RS7','White,Silver,Black,Red,Blue','petrol',73.0,'BS6','Luxury',8.9,25267789,875539,6,0)")
vehCur.execute("INSERT INTO Cars VALUES(32,'Audi
Q8','White,Silver,Black,Red,Orange','petrol',85.0,'BS6','SUV',9.8,11684890,419500,5,0)")
vehCur.execute("INSERT INTO Cars VALUES(33,'Audi e-
tron','White,Silver,Black,Red,Blue','electric',0.0,'ZEV','SUV',9.8,10199990,0,4,0)")
con.commit()
```

#TO INPUT BANK DATA

```
vehCur.execute("CREATE TABLE bank(Sl_No int,BANK char(20) primary key,LOWER_LIMIT float,UPPER_LIMIT float)")
vehCur.execute("INSERT INTO bank VALUES(1,'ICICI Bank',8.15,20.00)")
vehCur.execute("INSERT INTO bank VALUES(2,'HDFC Bank',8.00,20.00)")
vehCur.execute("INSERT INTO bank VALUES(3,'SBI Bank',7.70,15.00)")
vehCur.execute("INSERT INTO bank VALUES(4,'Axis Bank',7.00,17.75)")
vehCur.execute("INSERT INTO bank VALUES(5,'CBI Bank',8.90,10.75)")
vehCur.execute("INSERT INTO bank VALUES(6,'Canara Bank',7.30,9.90)")
vehCur.execute("INSERT INTO bank VALUES(7,'Bank Of Baroda',7.00,15.00)")
vehCur.execute("INSERT INTO bank VALUES(8,'Federal Bank',8.50,15.00)")
vehCur.execute("INSERT INTO bank VALUES(9,'IDBI Bank',9.10,9.70)")
vehCur.execute("INSERT INTO bank VALUES(10,'Syndicate Bank',8.85,10.00)")
con.commit()
```

#TO INPUT ALLY COMPANIES

```
vehCur.execute("create table Ally_Companies(Sl_no int,Company_name char(30),No_of_Cars int,N0_of_Scooters int, No_of_Bikes int, Total_Products int)")
vehCur.execute("insert into Ally_Companies values(1,'Apache',0,0,2,2)")
vehCur.execute("insert into Ally_Companies values(2,'Ather',0,1,0,1)")
vehCur.execute("insert into Ally_Companies values(3,'Audi',3,0,0,3)")
vehCur.execute("insert into Ally_Companies values(4,'Bajaj',0,1,1,1)")
vehCur.execute("insert into Ally_Companies values(5,'BMW',2,1,1,4)")
vehCur.execute("insert into Ally_Companies values(6,'Duccati',0,0,1,1)")
vehCur.execute("insert into Ally_Companies values(7,'Harley Davidson',0,0,1,1)")
vehCur.execute("insert into Ally_Companies values(8,'Hero',0,2,0,2)")
vehCur.execute("insert into Ally_Companies values(9,'Honda',1,1,2,4)")
vehCur.execute("insert into Ally_Companies values(10,'Hyundai',3,0,0,3)")
vehCur.execute("insert into Ally_Companies values(11,'Kia',2,0,0,2)")
vehCur.execute("insert into Ally_Companies values(13,'Kawasaki',0,0,1,1)")
vehCur.execute("insert into Ally_Companies values(14,'Maruti',4,0,0,4)")
vehCur.execute("insert into Ally_Companies values(15,'Mahindra',2,0,0,2)")
vehCur.execute("insert into Ally_Companies values(16,'MG Hector',1,0,0,1)")
vehCur.execute("insert into Ally_Companies values(17,'Mercedes Benz',2,0,0,2)")
vehCur.execute("insert into Ally_Companies values(18,'Ola',0,1,0,1)")
vehCur.execute("insert into Ally_Companies values(19,'Renault',1,0,0,1)")
vehCur.execute("insert into Ally_Companies values(20,'Royal Enfield',0,0,2,2)")
vehCur.execute("insert into Ally_Companies values(21,'Rolls Royce',3,0,0,3)")
vehCur.execute("insert into Ally_Companies values(22,'Simple One',0,1,0,1)")
vehCur.execute("insert into Ally_Companies values(23,'Suzuki',0,1,1,2)")
vehCur.execute("insert into Ally_Companies values(24,'Tata',4,0,0,4)")
vehCur.execute("insert into Ally_Companies values(25,'Toyota',2,0,0,2)")
vehCur.execute("insert into Ally_Companies values(26,'TVS',0,1,1,2)")
con.commit()
```

#TO CREATE CUSTOMERS TABLE

```
vehCur.execute("create table Customers(Customer_ID int primary key,Name varchar(30),Phone_No bigint,Date_Of_Purchase date,Vehicle_Name varchar(20), Vehicle_Cost int,EMI_amountpermonth int,Final_Amount int)")
```

#TO CREATE EMPLOYEE TABLE

```
vehCur.execute("CREATE TABLE Employee(Sl_No int,EMPLOYEE_ID char(30),FIRST_NAME char(20),SURNAME char(20),AGE int,YEAR_OF_JOINING int,POSITION char(20),BASIC_PAY float,DEARANCE_ALLOWANCE float,TRAVEL_ALLOWANCE float,HRA float,INCOME_TAX float,PF float,NET_PAY float,NO_OF_VEHICLES_SOLD float)")
```

#FUNCTIONS

#Code for inputting employee details

```
SlNo=1        #Serial Number for Employee Table
def employee():
    global SlNo
    while True:
      print()
      import mysql.connector
      con=mysql.connector.connect(host="localhost",user='root',passwd='',database='project')
      vehCur=con.cursor()
      eId=input("Enter Employee ID: ")
      fn=input("Enter First Name of the Employee: ")
      sn=input("Enter Surname of the Employee: ")
      age=int(input("Enter the age of the Employee: "))
      yoj=int(input("Enter the year the Employee joined the company: "))
      p=False
      bp=0
      da=0
      ta=0
      hra=0
      pf=0
      it=0
      np=0
      while p==False:
         po=input("Enter the position of the Employee: ")
         p=True
         if(po.upper()=="SALES MANAGER"):
            bp=60000
            da=0.20*bp
            ta=0.10*bp
            hra=0.30*bp
            pf=0.15*bp
            it=0.20*bp
         elif(po.upper()=="ASSISTANT MANAGER"):
            bp=55000
            da=0.18*bp
            ta=0.12*bp
            hra=0.25*bp
            pf=0.13*bp
            it=0.20*bp
         elif(po.upper()=="SALESPERSON"):
            bp=56000
            da=0.15*bp
            ta=0.15*bp
            hra=0.23*bp
            pf=0.12*bp
            it=0.20*bp
            elif(po.upper()=="SECURITY GUARD"):
                bp=27500
                da=0.13*bp
                ta=0.15*bp
                hra=0.20*bp
                pf=0.12*bp
                it=0.30*bp
            elif(po.upper()=="ACCOUNTANT"):
                bp=50000
                da=0.17*bp
                ta=0.10*bp
                hra=0.24*bp
                pf=0.13*bp
                it=0.30*bp
            elif(po.upper()=="RECEPTIONIST"):
                bp=42000
                da=0.15*bp
                ta=0.12*bp
                hra=0.23*bp
                pf=0.13*bp
                it=0.30*bp
            elif(po.upper()=="SWEEPER"):
                bp=24000
                da=0.13*bp
                ta=0.15*bp
                hra=0.20*bp
                pf=0.12*bp
                it=0.30*bp
            elif(po.upper()=="MECHANIC"):
                bp=45000
                da=0.10*bp
                ta=0.15*bp
                hra=0.22*bp
                pf=0.13*bp
                it=0.30*bp
            else:
                print("Wrong Input.Please try again")
                p=False
        np=bp+da+ta+hra-pf-it
        vs=int(input("Enter no of vehicles sold till date: "))
        dat="INSERT INTO Employee
VALUES("+str(SlNo)+",'"+eId+"','"+fn+"','"+sn+"',"+str(age)+","+str(yoj)+",'"+po+"',"+str(bp)+","+str(
da)+","+str(ta)+","+str(hra)+","+str(pf)+","+str(it)+","+str(np)+","+str(vs)+")"
        vehCur.execute(dat)
        con.commit()
        SlNo+=1
        ch=input("Do you want to enter details of more employees?(Y/N): ")
        if(ch=="N" or ch=="n"):
            break
```

```
#AGENCY POV

def agency_pov():
 con=mysql.connector.connect(host="localhost",user='root',passwd=",database='project')
 vehCur=con.cursor()
 print("-----------------------------WELCOME TO PHOENIX AUTOMOBILES' COMPANY PORTAL---
--------------------------")
 while True:
    print()
    print()
    print("PLEASE CHOOSE THE ACTION TO BE CARRIED OUT.")
    print("  1.To add Employee Details.")
    print("  2.To view Employee Details.")
    print("  3.To view Customer Details.")
    print("  4.To view Number of Vehicles sold so far.")
    print("  5.To view Ally Company Details.")
    print("  6.To view Amount Collected so far with the Biggest Sale Of the Day")
    print("  7.Exit Agency POV")
    cho=int(input("PLEASE ENTER OPTION NUMBER: "))
    print()
    if(cho==1):
        employee()
    elif(cho==2):
        vehCur.execute("SELECT * FROM Employee")
        det=vehCur.fetchall()
        if(len(det)==0):
            print("No Employee Details found")
        else:
            for i in det:
                print()
                print("                        Details of Employee No.",i[0],":")
                print()
                print("Employee ID:",i[1])
                print("First Name:",i[2])
                print("Surname:",i[3])
                print("Age:",i[4])
                print("Year of Joining:",i[5])
                print("Position in Office:",i[6])
                print("Number of vehicles sold:",i[14])
                print("Basic Pay:",i[7])
                print("Dearance Allowance:",i[8])
                print("Travel Allowance:",i[9])
                print("House Rent Allowance:",i[10])
                print("PF:",i[11])
                print("Income Tax:",i[12])
                print("Net Salary:",i[13])
    elif(cho==3):
        vehCur.execute("SELECT * FROM Customers")
        det=vehCur.fetchall()
        if(len(det)==0):
            print("No Customer Details found")
        else:
            for i in det:
                print("DETAILS OF CUSTOMER WITH CUSTOMER ID",i[0])
                print()
                print("Customer Name:",i[1])
                print("Phone Number:",i[2])
                print("Date of Purchase:",i[3])
                print("Vehicle Purchased:",i[4])
                print("Cost of",i[4],":",i[5])
                if(i[6]==0):
                    print("EMI option not selected!")
                else:
                    print("EMI per month:",i[6])
                print("Final Amount:",i[7])
                print()
        print()

    elif(cho==4):
        print()
        print("No of Cars sold:",noc)
        print("No of Scooters sold:",nos)
        print("No of Bikes sold:",nob)
        print()
    elif(cho==5):
        print()
        vehCur.execute("SELECT * FROM Ally_Companies")
        dat=vehCur.fetchall()
        for i in dat:
            print("Company No.",i[0])
            print("Name:",i[1])
            print("Number of Cars:",i[2])
            print("Number of Bikes:",i[4])
            print("Number of Scooters:",i[3])
            print()
    elif(cho==6):
        print()
        print("Amount Collected:",amnt)
        print("Biggest sale of the day:",bsod)
        print()
    elif(cho==7):
        break
    else:
        print("Wrong input! Please try again!")
```

```
#COMPARISON
nob=0
noc=0
nos=0
amnt=0
bsod=0
def Comparison(v):
    global noc
    global nob
    global nos
    global Vehicle
    global Vehicledata
    global Vc
    import mysql.connector
    Vcon=mysql.connector.connect(host="localhost",user="root",passwd="",database="Project")

    Vcur=Vcon.cursor()
    if v==1:
      b="Yes"
      while b=="Yes":
        Vcur.execute("select MODEL from bikes")
        bNames=Vcur.fetchall()
        print("Select from:")
        blist=[]
        for bMname in bNames:
          print(bMname[0])
          blist.append(bMname[0])
        print()
        bn1=input("Enter name of first bike:")
        bn2=input("Enter name of second bike:")
        print()

        if bn1 and bn2 in blist:
          print("Sl_No\tMODEL\tCOLOURS_AVAILABLE\t\t\tFUEL_TYPE\t
BODY_TYPE\tMILEAGE(kmpl)\tTRANSMISSION\tENGINE_TYPE\tFUEL_TANK_CAPACITY(L)
\tNo_Of_CYLINDERS\tKERB_WEIGHT\tDISPLACEMENT\tEMISSION_STANDARD\tPRICE\tW
ARRANTY\tSTOCK_LEFT")
          Vcur.execute("select * from bikes where MODEL='"+bn1+"'")
          Datab1=Vcur.fetchone()
          for bval1 in Datab1:
            print(bval1,end='\t')
          print()
          Vcur.execute("select * from bikes where MODEL='"+bn2+"'")
          Datab2=Vcur.fetchone()
          for bval2 in Datab2:
            print(bval2,end='\t')
          print()
          print()
          b=input("Do you want to compare again?(Yes/No):")
```

```
                else:
                    print("Model Name entered is wrong. Kindly enter again")
                    b="Yes"
            print("ENTER THE MODEL NAME YOU HAVE CHOSEN")
            Vehicle=input("[Make sure you have chosen the vehicle you wish to buy]:")
            Vcur.execute("select * from bikes where MODEL='"+Vehicle+"'")
            Vehicledata=Vcur.fetchone()
            Vc=Vehicledata[13]
            nob+=1

        elif v==2:
            s="Yes"
            while s=="Yes":
                Vcur.execute("select MODEL from scooter")
                sNames=Vcur.fetchall()
                print("Select from:")
                slist=[]
                for sMname in sNames:
                    print(sMname[0])
                    slist.append(sMname[0])
                print()
                sn1=input("Enter name of first scooter:")
                sn2=input("Enter name of second scooter:")
                print()

                if sn1 and sn2 in slist:
                    print("Sl_No\tMODEL\tCOLOURS_AVAILABLE\t\t\tENGINE_TYPE\t
RANGE_ESCOOTER\tMOTOR_POWER\tCHARGING_TIME\tBATTERY_CAPACITY\tCHASSIS\t
TOP_SPEED\tMILEAGE\tDISPLACEMENT\tEMISSION_STANDARD\tFUEL_TANK_CAPACITY
\tLOAD_CAPACITY\tPRICE\tWARRANTY\tSTOCK_LEFT" )
                    Vcur.execute("select * from scooter where MODEL='"+sn1+"'")
                    Datas1=Vcur.fetchone()
                    for sval1 in Datas1:
                        print(sval1,end='\t')
                    print()
                    Vcur.execute("select * from scooter where MODEL='"+sn2+"'")
                    Datas2=Vcur.fetchone()
                    for sval2 in Datas2:
                        print(sval2,end='\t')
                    print()
                    print()
                    s=input("Do you want to compare again?(Yes/No):")
                else:
                    print("Model Name entered is wrong. Kindly enter again")
                    s="Yes"
            print("ENTER THE MODEL NAME YOU HAVE CHOSEN")
            Vehicle=input("[Make sure you have chosen the vehicle you wish to buy]:")
            Vcur.execute("select * from scooter where MODEL='"+Vehicle+"'")
            Vehicledata=Vcur.fetchone()
            Vc=Vehicledata[15]
            nos+=1

        elif v==3:
            c="Yes"
            while c=="Yes":
                Vcur.execute("select MODEL from Cars")
                cNames=Vcur.fetchall()
                print("Select from:")
                clist=[]
                for cMname in cNames:
                    print(cMname[0])
                    clist.append(cMname[0])
                print()
                cn1=input("Enter name of first Car:")
                cn2=input("Enter name of second Car:")
                print()

                if cn1 and cn2 in clist:
                    print("Sl_No\tMODEL\tCOLOURS_AVAILABLE\t\t\tFUEL_TYPE\t
FUEL_TANK_CAPACITY_Litres\tEMISSION_STANDARD\tBODY_TYPE\tMILEAGE_kmpl\tPRI
CE\tINSURANCE\tWARRANTY\tSTOCK_LEFT\t" )
                    Vcur.execute("select * from Cars where MODEL='"+cn1+"'")
                    Datac1=Vcur.fetchone()
                    for cval1 in Datac1:
                        print(cval1,end='\t')
                    print()
                    Vcur.execute("select * from Cars where MODEL='"+cn2+"'")
                    Datac2=Vcur.fetchone()
                    for cval2 in Datac2:
                        print(cval2,end='\t')
                    print()
                    print()
                    c=input("Do you want to compare again?(Yes/No):")
                else:
                    print("Model Name entered is wrong. Kindly enter again")
                    c="Yes"
            print("ENTER THE MODEL NAME YOU HAVE CHOSEN")
            Vehicle=input("[Make sure you have chosen the vehicle you wish to buy]:")
            Vcur.execute("select * from Cars where MODEL='"+Vehicle+"'")
            Vehicledata=Vcur.fetchone()
            Vc=Vehicledata[8]
            noc+=1
        print()
        print("DETAILS of",Vehicle,"\n",Vehicledata)
        Vcon.close()
```

```
#TO INPUT CUSTOMER DATA
SlNO=1
def Customers(N,Phone,date,VehiN,Vcost,emi,fa):
        import mysql.connector as my
        mycon=my.connect(host="localhost",user="root",passwd="",database="Project")
        mycur=mycon.cursor()
        global SlNO
        mycur.execute("INSERT INTO Customers
VALUES("+str(SlNO)+",'"+N+"',"+str(Phone)+",'"+str(date)+"','"+VehiN+"',"+str(Vcost)+","+str(emi)
+","+str(fa)+")")
        mycon.commit()
        SlNO+=1
        mycon.close()
```

```
#TO UPDATE AGENT DETAILS
def updateagent():
        import mysql.connector
        UAcon=mysql.connector.connect(host="localhost",database='Project',user='root',passwd='')
        UAcur=UAcon.cursor()
        AgentID=input("Enter Agent ID:")
        UAcur.execute("select NO_OF_VEHICLES_SOLD from Employee where
EMPLOYEE_ID="+str(AgentID))
        NO_of_Vehiold=int(UAcur.fetchone()[0])
        UAcur.execute("update Employee set NO_OF_VEHICLES_SOLD="+str(NO_of_Vehiold+1)+"
where EMPLOYEE_ID="+str(AgentID))
        UAcon.commit()
        UAcon.close()
```

```
#EMI
def EMIfunc(amount):
        import  mysql.connector as slt
        EMICon=slt.connect(host='localhost',user='root',password='',database='project')
        EMICur=EMICon.cursor()
        EMICur.execute("select * from bank")
        Data=EMICur.fetchall()
        for i in Data:
                print('Bank =',i[1],end='\t')
                print('Lower Limit =',i[2],end='\t')
                print('Upper Limit =',i[3],end='\t')
                print()
        global Bank
        Bank=input('Enter the preferred bank:')
        rate=float(input('Enter the preferred rate:'))
        Rate=rate/100
        global time
        time=int(input('Enter the no.of months: '))
        x=(amount*Rate/12)*(1+Rate/12)**time
        y=((1+Rate/12)**time)-1
        emi=x/y
```

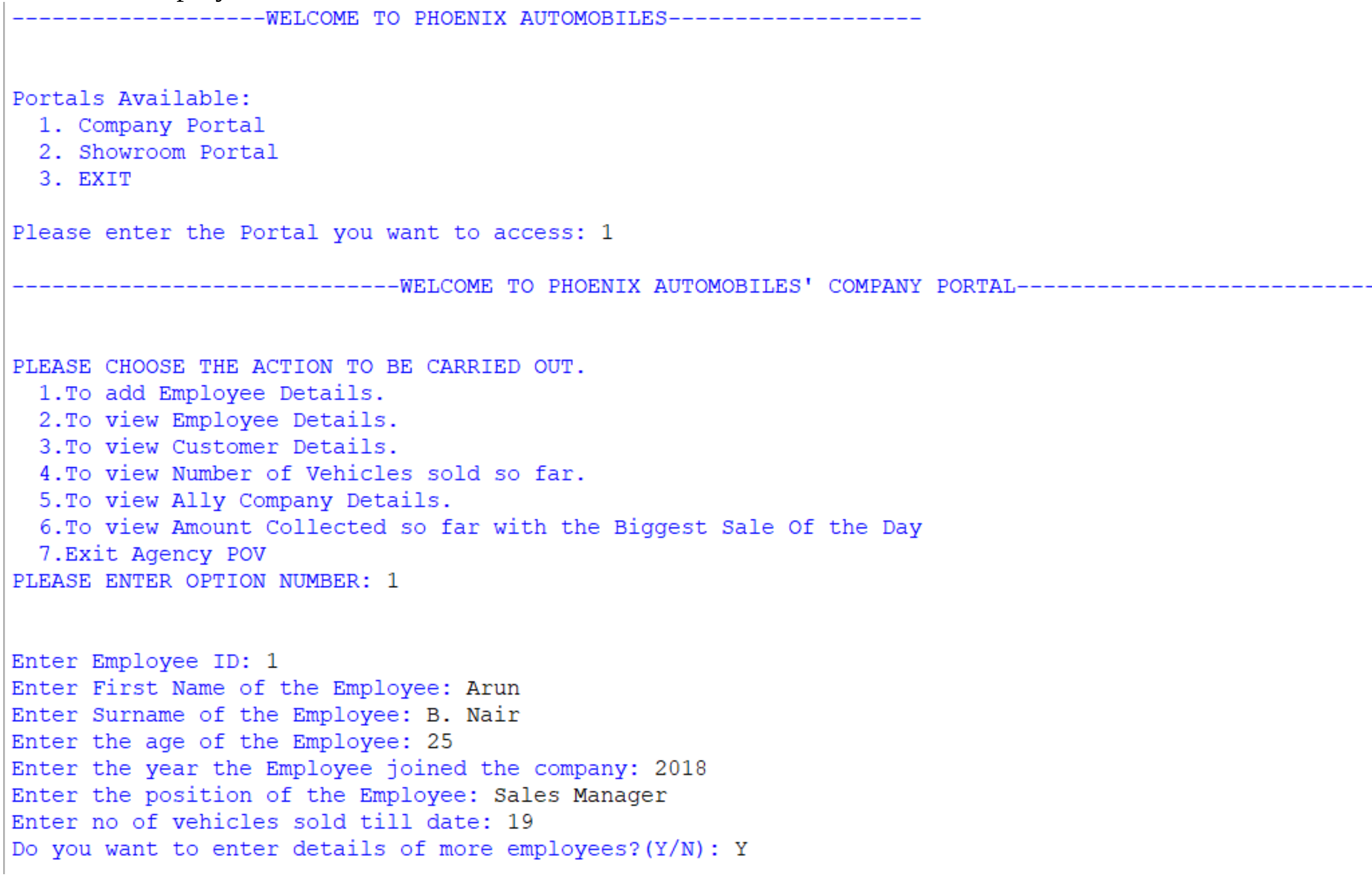
```
    EMICon.commit()
    EMICon.close()
    return emi


#CUSTOMER POV

def Customerpov():
    print("----------------------------WELCOME TO PHOENIX AUTOMOBILES' SHOWROOM PORTAL----------------------------")
    print("Enter 1 for Bikes")
    print("Enter 2 for Scooters")
    print("Enter 3 for Cars")
    vehiclechoice=int(input("Enter your choice:"))

    Comparison(vehiclechoice)
    print()
    global Vehicle
    global Vc
    global amnt
    global bsod

    Vn=Vehicle
    print("TO CHOOSE BANK")
    EMI=EMIfunc(Vc)
    global Bank
    global time
    FA=EMI*time
    amnt+=FA
    if FA>=bsod:
        bsod=FA

    print()
    updateagent()

    print()
    print("ENTER YOUR DETAILS")
    NM=input("Enter the name of Customer:")
    Ph=int(input("Enter the phone number of the Customer:"))
    import mysql.connector
    mycon=mysql.connector.connect(host="localhost",database='Project',user='root',passwd='')
    mycur=mycon.cursor()
    mycur.execute("select curdate()")
    dt=mycur.fetchone()[0]
    Customers(NM,Ph,dt,Vn,Vc,EMI,FA)
    print()
    print("YOUR ORDER HAS BEEN PLACED")
    print()
```

```
        print("--------------------------------BILLING---------------------------------")
        print("Name                  :",NM)
        print("Vehicle Name          :",Vn)
        print("Contact no.           :",Ph)
        print("Date of Purchase      :",dt)
        print("Price of Vehicle      :",Vc)
        print("EMI(amount per month)  :",EMI)
        print("EMI payment for(months):",time)
        print("Bank                  :",Bank)
        print("Total Amount          :",FA)

        print()
        print("---------------WE THANK YOU FOR YOUR CORRESPONDENCE WITH US---------------
")
        print()
        print()
        mycon.close()

#MAIN PROGRAM

while True:
    print("-------------------WELCOME TO PHOENIX AUTOMOBILES-------------------")
    print()
    print()
    print("Portals Available:")
    print("  1. Company Portal")
    print("  2. Showroom Portal")
    print("  3. EXIT")
    print()
    op=int(input("Please enter the Portal you want to access: "))
    print()
    if(op==1):
        agency_pov()
    elif(op==2):
        Customerpov()
    else:
        con.close()
        break
```

OUTPUT

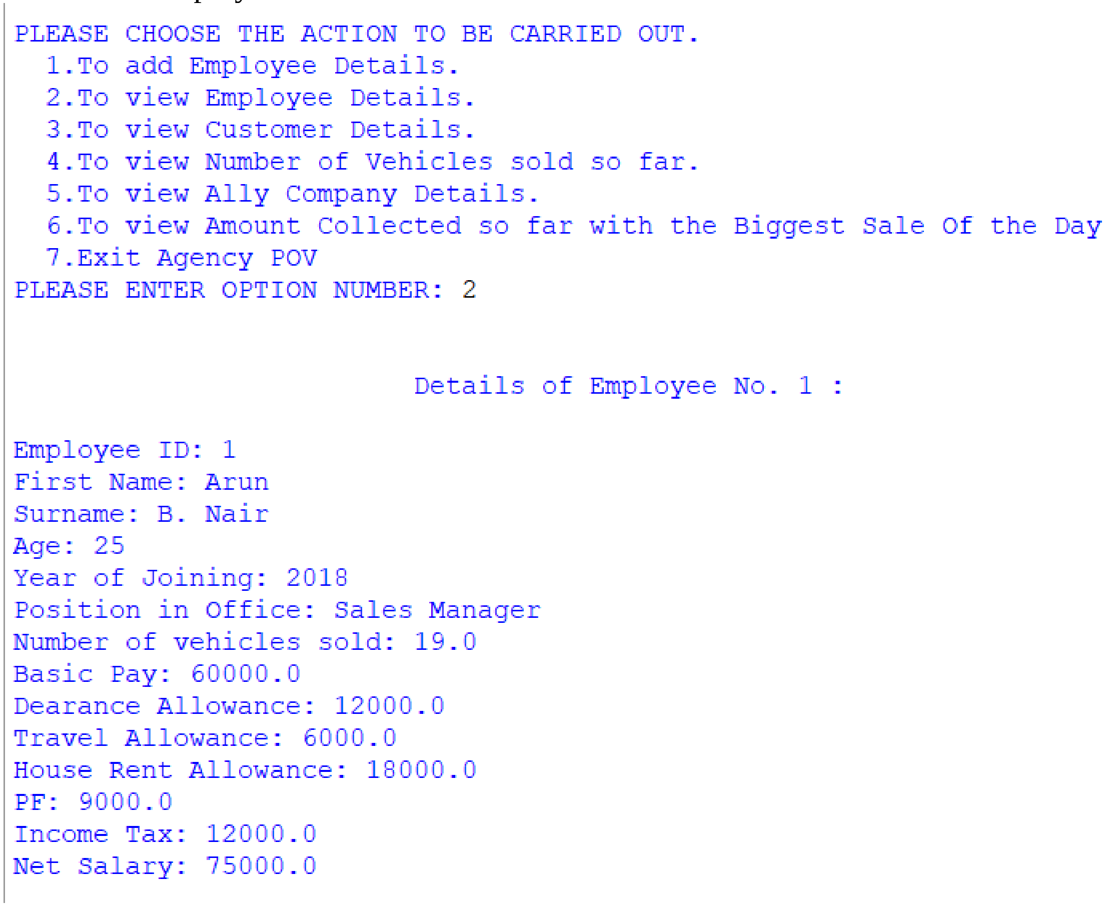
EMPLOYEE POINT OF VIEW:

1. To add Employee details



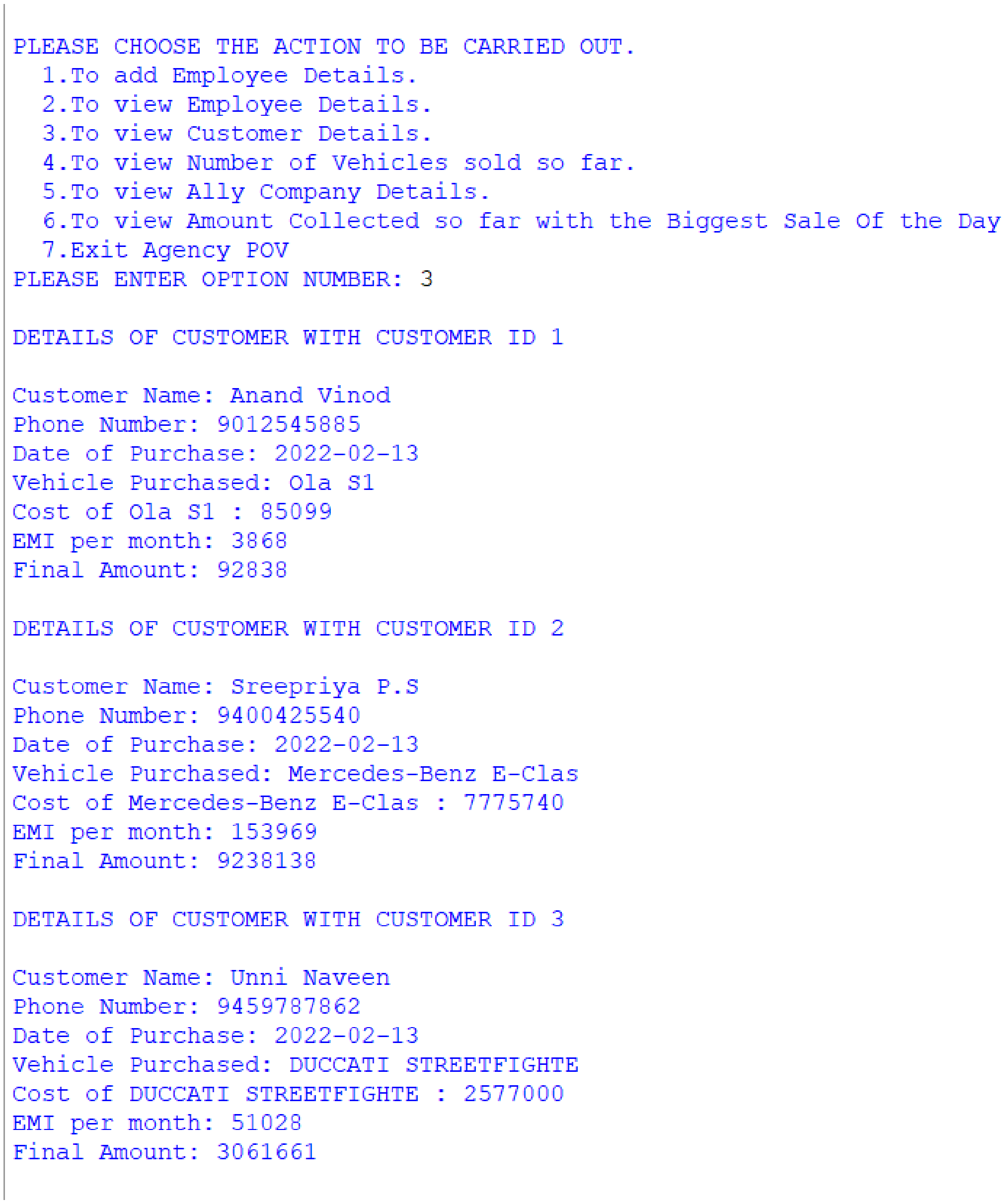
```
-------------------WELCOME TO PHOENIX AUTOMOBILES-------------------


Portals Available:
  1. Company Portal
  2. Showroom Portal
  3. EXIT

Please enter the Portal you want to access: 1

-----------------------------WELCOME TO PHOENIX AUTOMOBILES' COMPANY PORTAL---------------------------


PLEASE CHOOSE THE ACTION TO BE CARRIED OUT.
  1.To add Employee Details.
  2.To view Employee Details.
  3.To view Customer Details.
  4.To view Number of Vehicles sold so far.
  5.To view Ally Company Details.
  6.To view Amount Collected so far with the Biggest Sale Of the Day
  7.Exit Agency POV
PLEASE ENTER OPTION NUMBER: 1


Enter Employee ID: 1
Enter First Name of the Employee: Arun
Enter Surname of the Employee: B. Nair
Enter the age of the Employee: 25
Enter the year the Employee joined the company: 2018
Enter the position of the Employee: Sales Manager
Enter no of vehicles sold till date: 19
Do you want to enter details of more employees?(Y/N): Y
```

2. To view Employee Details



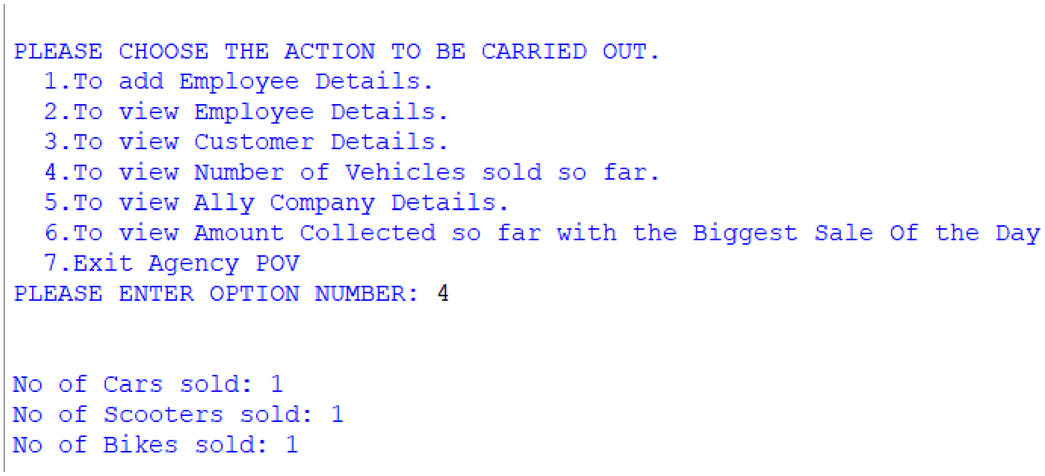
```
PLEASE CHOOSE THE ACTION TO BE CARRIED OUT.
  1.To add Employee Details.
  2.To view Employee Details.
  3.To view Customer Details.
  4.To view Number of Vehicles sold so far.
  5.To view Ally Company Details.
  6.To view Amount Collected so far with the Biggest Sale Of the Day
  7.Exit Agency POV
PLEASE ENTER OPTION NUMBER: 2


                         Details of Employee No. 1 :

Employee ID: 1
First Name: Arun
Surname: B. Nair
Age: 25
Year of Joining: 2018
Position in Office: Sales Manager
Number of vehicles sold: 19.0
Basic Pay: 60000.0
Dearance Allowance: 12000.0
Travel Allowance: 6000.0
House Rent Allowance: 18000.0
PF: 9000.0
Income Tax: 12000.0
Net Salary: 75000.0
```

3. To view Customer Details



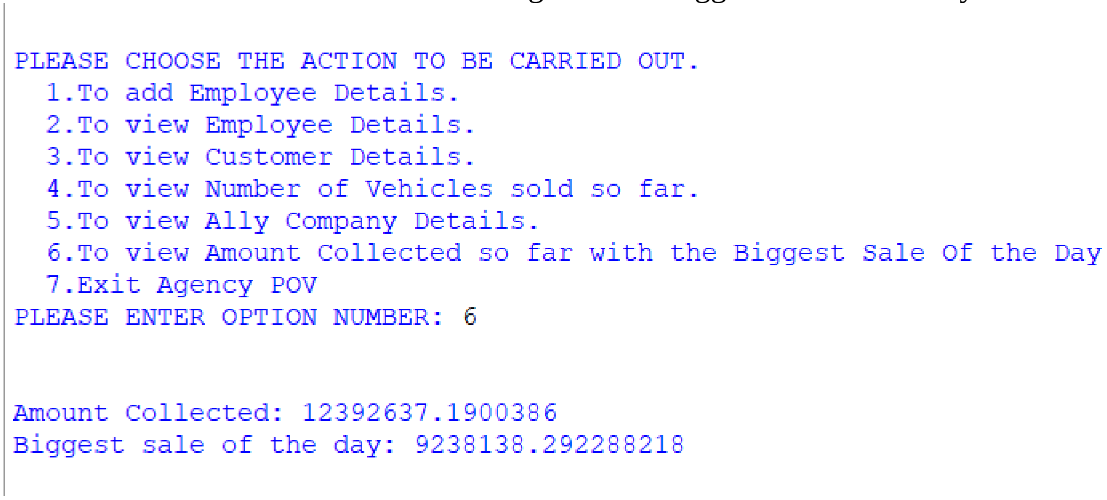
```
PLEASE CHOOSE THE ACTION TO BE CARRIED OUT.
  1.To add Employee Details.
  2.To view Employee Details.
  3.To view Customer Details.
  4.To view Number of Vehicles sold so far.
  5.To view Ally Company Details.
  6.To view Amount Collected so far with the Biggest Sale Of the Day
  7.Exit Agency POV
PLEASE ENTER OPTION NUMBER: 3

DETAILS OF CUSTOMER WITH CUSTOMER ID 1

Customer Name: Anand Vinod
Phone Number: 9012545885
Date of Purchase: 2022-02-13
Vehicle Purchased: Ola S1
Cost of Ola S1 : 85099
EMI per month: 3868
Final Amount: 92838

DETAILS OF CUSTOMER WITH CUSTOMER ID 2

Customer Name: Sreepriya P.S
Phone Number: 9400425540
Date of Purchase: 2022-02-13
Vehicle Purchased: Mercedes-Benz E-Clas
Cost of Mercedes-Benz E-Clas : 7775740
EMI per month: 153969
Final Amount: 9238138

DETAILS OF CUSTOMER WITH CUSTOMER ID 3

Customer Name: Unni Naveen
Phone Number: 9459787862
Date of Purchase: 2022-02-13
Vehicle Purchased: DUCCATI STREETFIGHTE
Cost of DUCCATI STREETFIGHTE : 2577000
EMI per month: 51028
Final Amount: 3061661
```

4. To view Number of Vehicles sold so far



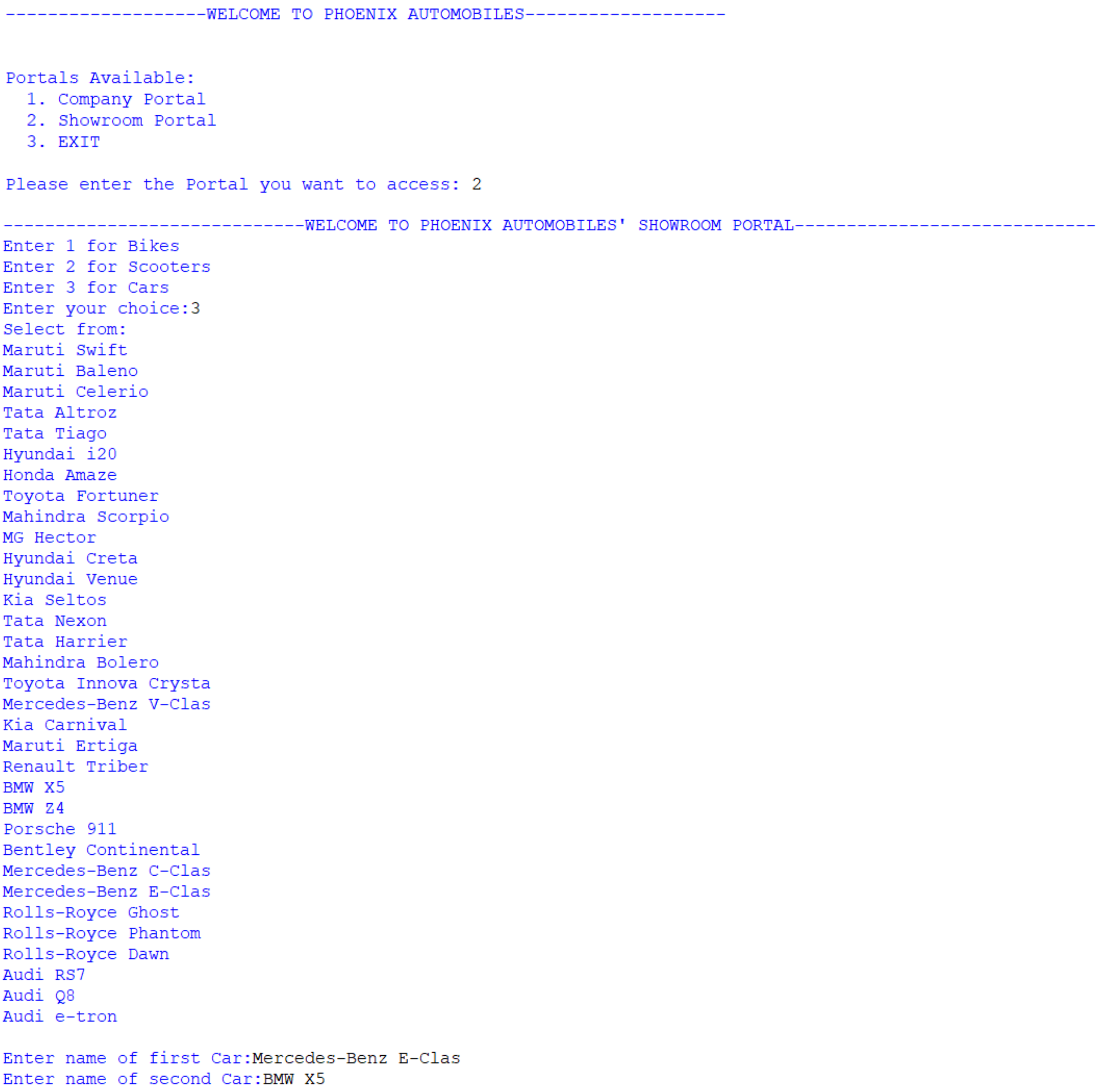
```
PLEASE CHOOSE THE ACTION TO BE CARRIED OUT.
  1.To add Employee Details.
  2.To view Employee Details.
  3.To view Customer Details.
  4.To view Number of Vehicles sold so far.
  5.To view Ally Company Details.
  6.To view Amount Collected so far with the Biggest Sale Of the Day
  7.Exit Agency POV
PLEASE ENTER OPTION NUMBER: 4


No of Cars sold: 1
No of Scooters sold: 1
No of Bikes sold: 1
```

6. To view Amount Collected so far along with the Biggest Sale Of the Day

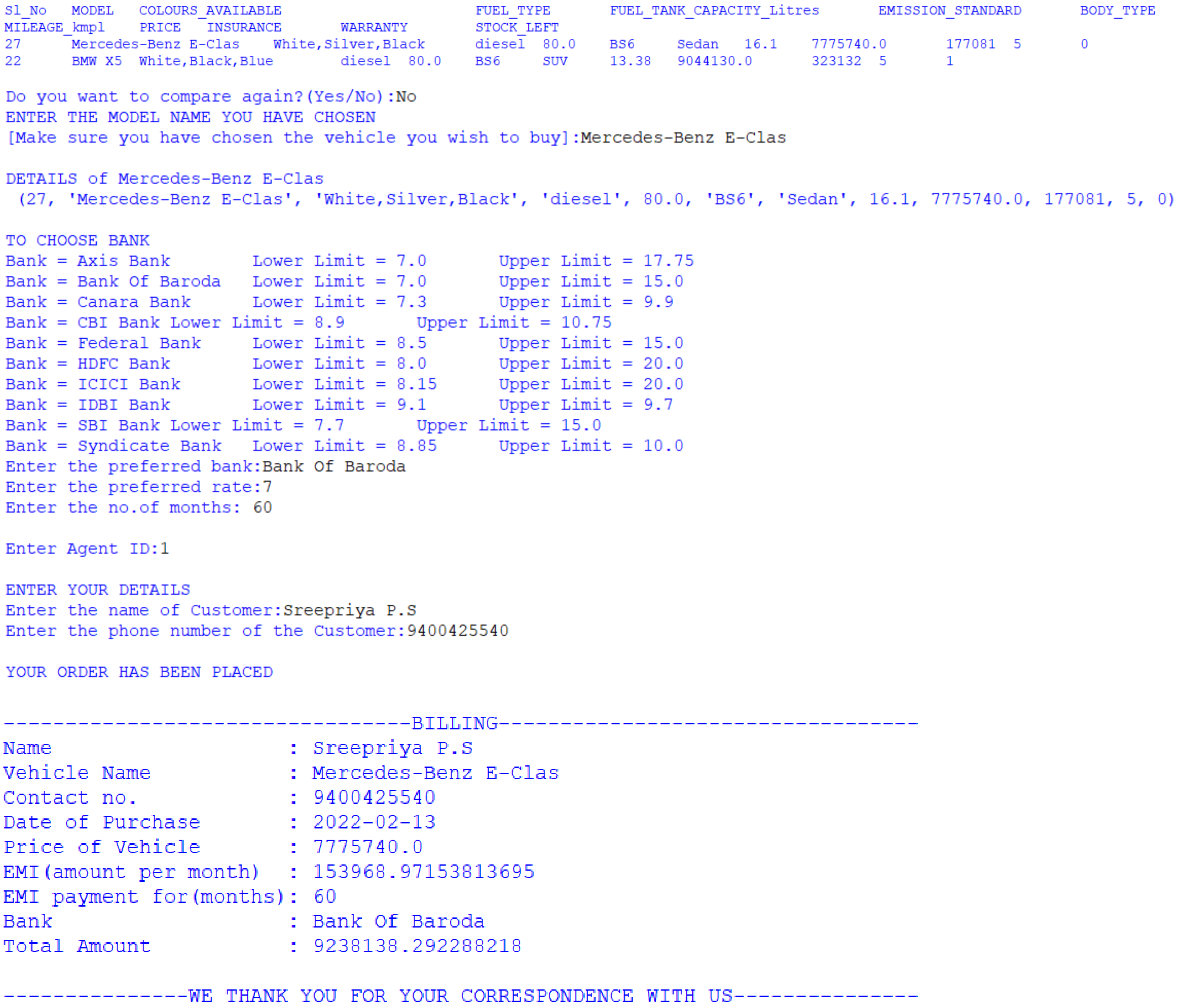


```
PLEASE CHOOSE THE ACTION TO BE CARRIED OUT.
  1.To add Employee Details.
  2.To view Employee Details.
  3.To view Customer Details.
  4.To view Number of Vehicles sold so far.
  5.To view Ally Company Details.
  6.To view Amount Collected so far with the Biggest Sale Of the Day
  7.Exit Agency POV
PLEASE ENTER OPTION NUMBER: 6


Amount Collected: 12392637.1900386
Biggest sale of the day: 9238138.292288218
```

CUSTOMER POINT OF VIEW



```
-------------------WELCOME TO PHOENIX AUTOMOBILES-------------------


Portals Available:
  1. Company Portal
  2. Showroom Portal
  3. EXIT

Please enter the Portal you want to access: 2

-----------------------------WELCOME TO PHOENIX AUTOMOBILES' SHOWROOM PORTAL-----------------------------
Enter 1 for Bikes
Enter 2 for Scooters
Enter 3 for Cars
Enter your choice:3
Select from:
Maruti Swift
Maruti Baleno
Maruti Celerio
Tata Altroz
Tata Tiago
Hyundai i20
Honda Amaze
Toyota Fortuner
Mahindra Scorpio
MG Hector
Hyundai Creta
Hyundai Venue
Kia Seltos
Tata Nexon
Tata Harrier
Mahindra Bolero
Toyota Innova Crysta
Mercedes-Benz V-Clas
Kia Carnival
Maruti Ertiga
Renault Triber
BMW X5
BMW Z4
Porsche 911
Bentley Continental
Mercedes-Benz C-Clas
Mercedes-Benz E-Clas
Rolls-Royce Ghost
Rolls-Royce Phantom
Rolls-Royce Dawn
Audi RS7
Audi Q8
Audi e-tron

Enter name of first Car:Mercedes-Benz E-Clas
Enter name of second Car:BMW X5
```



```
Sl_No   MODEL   COLOURS_AVAILABLE                     FUEL_TYPE       FUEL_TANK_CAPACITY_Litres       EMISSION_STANDARD       BODY_TYPE
MILEAGE_kmpl    PRICE   INSURANCE       WARRANTY        STOCK_LEFT
27      Mercedes-Benz E-Clas    White,Silver,Black      diesel  80.0    BS6     Sedan   16.1    7775740.0       177081  5       0
22      BMW X5  White,Black,Blue        diesel  80.0    BS6     SUV     13.38   9044130.0       323132  5       1

Do you want to compare again?(Yes/No):No
ENTER THE MODEL NAME YOU HAVE CHOSEN
[Make sure you have chosen the vehicle you wish to buy]:Mercedes-Benz E-Clas

DETAILS of Mercedes-Benz E-Clas
 (27, 'Mercedes-Benz E-Clas', 'White,Silver,Black', 'diesel', 80.0, 'BS6', 'Sedan', 16.1, 7775740.0, 177081, 5, 0)

TO CHOOSE BANK
Bank = Axis Bank        Lower Limit = 7.0       Upper Limit = 17.75
Bank = Bank Of Baroda   Lower Limit = 7.0       Upper Limit = 15.0
Bank = Canara Bank      Lower Limit = 7.3       Upper Limit = 9.9
Bank = CBI Bank Lower Limit = 8.9       Upper Limit = 10.75
Bank = Federal Bank     Lower Limit = 8.5       Upper Limit = 15.0
Bank = HDFC Bank        Lower Limit = 8.0       Upper Limit = 20.0
Bank = ICICI Bank       Lower Limit = 8.15      Upper Limit = 20.0
Bank = IDBI Bank        Lower Limit = 9.1       Upper Limit = 9.7
Bank = SBI Bank Lower Limit = 7.7       Upper Limit = 15.0
Bank = Syndicate Bank   Lower Limit = 8.85      Upper Limit = 10.0
Enter the preferred bank:Bank Of Baroda
Enter the preferred rate:7
Enter the no.of months: 60

Enter Agent ID:1

ENTER YOUR DETAILS
Enter the name of Customer:Sreepriya P.S
Enter the phone number of the Customer:9400425540

YOUR ORDER HAS BEEN PLACED

---------------------------------BILLING---------------------------------
Name                    : Sreepriya P.S
Vehicle Name            : Mercedes-Benz E-Clas
Contact no.             : 9400425540
Date of Purchase        : 2022-02-13
Price of Vehicle        : 7775740.0
EMI(amount per month)   : 153968.97153813695
EMI payment for(months) : 60
Bank                    : Bank Of Baroda
Total Amount            : 9238138.292288218

---------------WE THANK YOU FOR YOUR CORRESPONDENCE WITH US---------------
```

ADVANTAGES

- The project contains the code for the insertion of information of Vehicles(Bikes, Scooters and Cars), Banks, Ally Companies, Employees and Customers.
- The showroom agents can easily add and view employee details.
- The showroom agents can also view the Customer details.
- The number of vehicles sold so far, total amount collected and the biggest sale of the day can also be printed.
- Customers can easily compare between 2 vehicles.
- Customers can choose a vehicle of their choice.
- EMI option is also provided. Customers can choose a bank along with rate and time for the Emi payment.
- Agent details(Employee table) and Customer table gets updated simultaneously as the billing process proceeds.
- Finally, the bill is printed along with necessary details.

SCOPE FOR IMPROVEMENT

- This program can deal with the sales of only one day.
- Discount offers could also have been included.
- More vehicles like trucks, pick-ups, autos, etc. could also have been included.
- Programs for adding more vehicles to the tables: Cars, Bikes and Scooters could have been included.
- Programs for placing order after the billing to the respective companies could have been included.
- Only 2 vehicles have been compared in the program. The customer can be made to select the desired number of vehicles to compare.
- Vehicles could have been compared on a basis (like price).
- The data after comparison is not printed in the form of a table and hence hard to understand. The data can be more organized and printed as a table.
- It is assumed that the customer will always go for payment via EMI.
- It is assumed that the customer will always enter the details correctly.
- It is assumed that the rate entered by the customer is within the rate the bank provides.

BIBLIOGRAPHY

❖ CAR DETAILS	https://www.cardekho.com/compare-cars
❖ BIKE and SCOOTER DETAILS	https://www.bikedekho.com/compare
❖ EMI FORMULA	https://emicalculator.net/
❖ BANK DETAILS	https://www.bankbazaar.com/
❖ COMPUTER SCIENCE with python- Textbook for Class XII – by Sumita Arora	